Circulation 1400

RENEWAL TIME for issues 7 thru 12.

It's time to renew your subscription if you haven't done so already. New rates are: $5.00 (U.S. Funds) for U.S. & Canada. $10.00 (U.S. Funds) for overseas (includes envelope). Please mark RENEWAL on your checks and envelopes.

KIM-1/650X USER NOTES is published bimonthly by:

Eric C. Rehnke editor/publisher
Brenda J. Rehnke business manager

All subscriptions and/or correspondence should be addressed:

KIM-1 650X USER NOTES
P.O. Box 33077
North Royalton, Ohio 44133

Our new phone number is (216) 237-5571.

Article Contributors Please Note

To alleviate possible typographical errors and grey hairs, all programs and article contributions should be originals (not copies) and typed, using single space with 8" wide columns, if at all possible.

To enable others to understand your programs as well as you do, a well documented source listing is a necessity. A good example of an adequately commented program is the "PLL SET" program in #5 page 3.

WRONG WAY

```
179C 8C 40 17 SEGS STY SAD - Store "T" in 1740.
```

RIGHT WAY

```
179C 8C 40 17 SEGS STY SAD - Turn on the segments.
```

Very long programs should include a hex dump in case space doesn't permit publishing the entire listing.

HELP WANTED

I'd like to print a list of "good guys" who would be willing to help other members by answering questions thru the mail about KIM hardware and/or software. These other members would be required to send you an S.A.S.E. with their query so you wouldn't get stuck with return postage. Let me know your specialty. Now's your chance.

MORE KIM DISTRIBUTORS

ABComputers, P.O. Box 104, Perkasie, Pa. 18944

DERRICK ELECTRONICS, 714 West Kenosha, Broken Arrow, Okla. 74012. (918) 251-9923

LOCAL USER GROUPS

Santa Barbara, Cal. area: John Eaton (805) 682-1895

Tulsa, Oklahoma area: Dan Bates Rt 7 Box 310, Claremore Okla. 74017

OOPS!!

Remember the RIVERSIDE ELECTRONICS Application Notes which were mentioned on pg. 1 of #5? Well, the prices have been changed. Here's the new prices:
MVM-1,2,3,4,5 (concerns the MVM 1024 video display) $1.00/set of five.

KIM1-1 (expanding the KIM) $1.00
KIM1-2 (KIM software for the MVM-1024) $3.00

MORE KIM SOFTWARE:

PYRAMID DATA SYSTEMS has announced immediate availability of an extended I/O monitor package, "XIM", for KIM. "XIM" resides in a little less than 1K of memory, and adds 17 commands (4 are user definable) to a terminal equipped KIM. The list of commands includes: Block move, Block search, Block compare, Hex load & dump, breakpoint processing, relative branch calculation, etc... A 45 page user manual includes a complete commented surce listing of "XIM" and includes instructions on relocation of the monitor from its present $2000 starting address if necessary. Documentation looks very good.

The price? $10.00 for the manual and paper tape or $12.00 for the manual and KIM cassette. (N.J. residents add 5% tax). Send S.A.S.E. for more info: PYRAMID DATA SYSTEMS, 6 Terrace Avenue, New Egypt, N.J. 08533.

MICRO-WARE LTD. now has an assembler, disassembler, text editor package (MICRO-ADE 6502) ready for distribution. Micro-ade resides in 4K of memory and includes a two pass assembler which can be user configured to operate with two cassette recorders with start/stop controls or one manually controlled cassette. The 56 page user manual contains the source listings for all I/O routines which should enable one to interface this package to any peripheral device. The user manual and KIM cassette or paper tape costs $25.00 and the source listing for the whole package is an additional $25.00. For more info send S.A.S.E. to MICRO-WARE LTD., 27 Firstbrooke Road, Toronto, Ontario, Canada, M4E 2L2. Another well documented package.

TINY BASIC TAPES used to be available from the Byte Shop #2 until production problems forced them to discontinue. Fortunately TB is still available on KIM cassettes from: Kenneth W. Ensele, 1337 Foster Rd., Napa, Ca. 94558.

Price for Tom Pittmans 2K Tiny Basic on KIM cassette, at either address $200 or $2000 (please specify), is $9.50 plus $1.00 for postage and handling. Terms are: cash with order and please allow 30 days for delivery.

KIM ENCLOSURE

I recently received a flyer announcing the availability of an enclosure for the basic KIM board. The 2-piece molded plastic box forms a sandwich (KIM's in the middle) with openings for the keyboard, display and edge connectors. looks very neat and functional. Get the flyer from: THE ENCLOSURES GROUP, 55 Stevenson St., San Francisco, Calif. 94105.

THE FIRST BOOK OF KIM will be ready for distribution at the end of August. Stan Ockers, Jim Butterfield and your editor put this book together with the idea of helping newcomers to our hobby to get up to speed on the KIM. (of course,the book's not just applicable to newcomers). The book includes a beginners guide to programming, several tutorials on hooking things up to KIM, and a large number of game and utility type programs. (many of which have not been published as of yet) The First Book of KIM is 180 pages long in an 8½x11 format. It is available for $9.00 (plus $.50 postage) from: ORB, P.O. Box 311, Argonne, ILL., 60439. Personal checks will have to clear the bank, so please send a cashiers check or money order in U.S. funds. Ill residents please add sales tax.

THE COMPLIMENTARY ISSUE OF KIM USER NOTES is no longer available. The more noteworthy sections of the issue will continue to be reprinted in upcoming issues. (see the A/D converter in issue #4 page 9.)

EDITORS NOTE: Tom has done a great job in showing us how to certify our tapes before we entrust them to the task of "remembering". I have since switched over to the "Radio Shack" 30 minute "medium priced" tapes after the high priced 60 minute tapes proved unsatisfactory in their "fast forward" access time. With "Hypertape" (formerly Supertape) now in constant use here, and the digital tape counter on my "SANKYO" cassette recorder, I can put lots of programs on a little tape. Does anyone know if reasonably priced 15 minute tapes exist? (7½ minutes/side).

USING THE KIM-1 AUDIO CASSETTE INTERFACE

by Tom Marchant
5123 Trumbull
Detroit, Mich. 48208

Once a program has been dumped to audio cassette, and the power has been turned off, can it be loaded back in? If not, why not? Is there a way to tell if the information can be recovered before destroying what is in memory, and how reliable can the cassette be expected to be?

I have had considerable trouble with the audio interface. The cassette deck that I use is an Advent model 201 stereo deck. Frequency response shouldn't be a problem, especially since the highest frequency recorded is only 3700 Hertz. Any recorder should be able to handle that. Noise shouldn't be too much of a problem either with a good deck. So what is the problem?

To see what the information recorded on the tape looks like, I wrote some routines to read the tape and show what comes in from the cassette on the Hex display. When it is looking for a SYN character, it constantly rotates the leftmost digit, displaying the bit pattern of the last byte read in. When a SYN is found, the data digits show the SYN character code '16' while the address digits show a count of SYN characters. When it sees an '*', it picks up the ID and stores it in the data display, then puts the starting address into the address display. For each byte of data read in, the address is incremented. If nothing is on the tape, the routines notice that fact, and go back to looking for a SYN.

Using these routines, and another to write out a steady stream of SYN characters, I found that my biggest problem was dropouts on the tape. I also found that by turning up the record volume, I was able to get better results. In my original attempts I paid attention to my VU meter. At first I recorded at a 0 VU level. When I played the tape back, the meter went right off the scale! So, I turned it down to -7 VU, at which point it played back at around 0. One tape worked fine most of the time, but another would almost never work. At this point, my record level control was at about 2 (out of 10). I found that by recording in stereo mode with data coming in on the B channel and the meter switched to the A channel (so as not to peg the needle), and with the record level turned up high, I managed to blast past the dropouts. I suspect that part of my problems stem from using a stereo deck, where the heads are less than half the width of mono heads.

I find that all of the problems that I have with my deck have to do with too little level coming in to trigger the phase-locked loop. I have, however, seen cases where the level was so high that the carrier frequencies punched their way through the PLL to appear as noise in the signal at the PIA.

How can a tape be tested for dropouts before it is used? One simple method would be to record a steady stream of SYN characters on tape, and look at the results when playing it back. If it is necessary to re-synchronize the display will start counting from zero again. The PIA pin, when read by the CPU, always has either a zero or a one on it. In my KIM, this pin is a 1 when nothing is coming in. So what happens if the dropout is in the middle of the lower frequency tone (which also appears as a 1)? Nothing! And if a tape is checked using SYN characters (or any other character for that matter, since every bit has both tones in it), better hope and pray that any dropouts that couldn't be seen because they were in the middle of a low frequency tone don't end up in a high frequency tone when good data is recorded!

The solution is to write a steady high frequency tone on the tape, then read it back, looking for a 1 on the input. If any are found, and if no record/playback levels can be found to get around the problem, the tape can still be used to record music, but forget about putting data on it! I am working on some fancier routines for checking a tape that will tell the number of errors found, as well as the length of the shortest and longest errors. Perhaps they will be ready for the meeting, but not in time for inclusion in this newsletter. I have found that a dropout of less than 600 microseconds long will trigger the PLL. And each bit is about 7.5 miliseconds long!

TESTING A TAPE

Two routines are included. Starting at 0000 is a routine to write a steady tone to the tape. It does this by repeatedly calling subroutine ONE. Note that the subroutine called ONE writes out a high frequency tone, which comes back in as a 0 on the PIA. This routine can be modified to write out a low frequency tone by changing the instruction at NEXT to JSR to subroutine ZRO instead of ONE (change location 000B from 9E to C4).

The second routine (starting at 0010) reads the tape back. It displays a 7 bit count of the number of errors found on the tape. When it sees a 1 on the PIA pin, it adds 1 to SAD, then when it sees a 0 again, it goes back to looking for a 1. The number of errors is displayed as a 7 bit binary number in the leftmost digit of the display. The low order bit is the top segment, and successive higher order bits are found going clockwise from there, with the high order bit in the middle. The important thing is just that the display changes. When a dropout is found, it often appears as several errors. That is, it sees a 1, then a 0, then a 1, then a 0, etc. If the display stays blank, you are getting a 0, and there are no errors. Errors can be induced by turning down the playback volume. If nothing is coming in, an error will be indicated as soon as the program is started. If it is necessary to record a low frequency tone and test it coming in, three instructions need to be changed. The BPL instructions at 0025 and 002F need to be changed to BMI (30), and the BMI at 002D needs to be changed to BPL (10). If it is desired to have a visual display of the duration of the errors, delete all code from BAD (0C2A), and insert a JMP (or equivalent) to GOOD at 002A.

Both of these routines are fully relocatable, and independant of each other. They may be loaded and executed at any location with no modification. Since I have started to test my tapes before using them, I have had no problems getting programs back from tape. And I am no longer afraid to turn off the KIM when I'm in the middle of working on some new programs. I can save them -- and get them back!

more...

TAPE CERTIFYING BTH -- TOM MARCHANT

```
CARD #  LOC   CODE        CARD
     2                    ;       ROUTINE TO WRITE OUT STEADY TONE TO AUDIO TAPE.
     3                    ;
     4                    SAD     =$17A0
     5                    SADD    =$17A1
     6                    SBD     =$17A2
     7                    SBDD    =$17A3
     8                    ONE     =$199E
     9                    ;
    10  0000              WRITE   *=0
    11  0000  A9 27       WRITE   LDA  #$27
    12  0002  8D 42 17             STA  SBD
    13  0005  A9 BF               LDA  #$BF
    14  0007  8D 43 17             STA  SBDD
    15  000A  20 9E 19     NEXT    JSR  ONE        MAKE TONE.
    16  000D  38                   SEC
    17  000E  B0 FA                BCS  NEXT       UNCONDITIONAL BR.
    18                    ;
    19                    ;
    20                    ;       ROUTINE TO READ BACK TAPE LOOKING FOR
    21                    ;       DROPOUTS.
    22                    ;
    23  0010  A9 00       LPE     LDA  #0
    24  0012  8D 40 17             STA  SAD
    25  0015  A9 7F               LDA  #$7F
    26  0017  8D 41 17             STA  SADD
    27  001A  8D 43 17             STA  SBDD
    28  001D  A9 09               LDA  #$09
    29  001F  8D 42 17             STA  SBD
    30  0022  2C 42 17     GOOD    BIT  SBD        ALL OK?
    31  0025  10 FB                BPL  GOOD       -YES- KEEP LOOKING.
    32  0027  EE 40 17             INC  SAD        MAKE LED BLINK.
    33  002A  2C 42 17     BAD     BIT  SBD        STILL BAD?
    34  002D  30 FB                BMI  BAD        -YES- SPAN DROPOUT.
    35  002F  10 F1                BPL  GOOD
    36                    ;
    37                    ;       HAPPY HUNTING.
    38                    .END
```

...and here's a handy time saver...

PROGRAM CYCLE COUNTER — C.H.Gould, 317 Cocoa, Indialantic FL 32903

I hate to count up the cycles in a program segment to make it come out right. Here is a simple cycle counter which displays in hexidecimal form the cycle (microsecond) length of a program or segment thereof. The segment cannot be longer than 256 (FF) cyles. Write starting address of program segment at 17C8 (LO) and 17C9 (HI). After last step in program segment to be tested, write 4C CA 17. Dont forget to remove later. To use, start at 17C0, and read cycle time on low bits of address display.

```
17C0 18        CLC
17C1 D8        CLD
17C2 A9 0A     LDA #$0A Set timer
17C4 8D 44 17  STA
17C7 4C -- --  JMP      to program
17CA A9 FF     LDA #$FF  Return
17CC ED 46 17  SBC      Read time
17CF 85 FA     STA      PointL
17D1 A9 00     LDA #$00
17D3 85 FB     STA      PointH
17D5 4C 4F 1C  JMP      Display cycles.
```

from:
Jack Babcock
1016 N Sage Ave.
Rialto, Calif 92376

After pumching in WUMPUS (works great) I decided there has to be an easier way, hence the enclosed program. It's still tedious, but the program helps some.

PROGRAM HANDLOADER takes a little of the pain out of handloading long programs by automatically stepping the address every two numeric keystrokes, and by providing an address backstep function.

The program is shown in memory locations 17A1-17E6, but is completely relocateable, and may reside in any 70 (46 hex) consecutive bytes of RAM.

After loading, access the program as follows:
(a)Manual load *A9 into 17FA (NMIL)
*17 into 17FB (NMIH)
(b)Set address to first memory to be loaded.
(c)Press the ST key.

You are now in the program, and the keys function as follows:

+ Increments address.
PC Decrements address.
AD, DA, & GO reset address toggle to 0. That is, 2 numeral keys will now be required before address is incremented.
RS Return to KIM-1 monitor (START).
0-F Shift in numerical data.

Note that the address does not increment until the numeric key is released, so that memory and contents may be checked by holding the key down.

The ease of transition between this program and the KIM-1 monitor(single keystroke, no address change,both ways) encourages the use of the features of both programs.

```
17A1  A5  FA        BSTEP   LDA  POINTL
17A3  D0  02                BNE  BSTEP1
17A5  C6  FB                DEC  POINTH
17A7  C6  FA        BSTEP1  DEC  POINTL
* 17A9  D8          HANDL   CLD              set hex mode
17AA  B8                    CLV              setup for relative jump
17AB  A2  03        HANDL1  LDX  #03         reset address toggle,
17AD  86  FD                STX  TMPX        get 2 keys
17AF  C6  FD        HANDL2  DEC  TMPX        per address
17B1  D0  0A                BNE  HANDL3
17B3  20  19  1F    WAIT    JSR  SCAND       on 2nd key
17B6  D0  FB                BNE  WAIT        release,
17B8  20  63  1F    STEP    JSR  INCPT       increment address
17BB  50  EE                BVC  HANDL1      relative jump
17BD  20  19  1F    HANDL3  JSR  SCAND       wait for
17C0  D0  FB                BNE  HANDL3      key release
17C2  20  19  1F    HANDL4  JSR  SCAND
17C5  F0  FB                BEQ  HANDL4      wait for key
17C7  20  19  1F            JSR  SCAND
17CA  F0  F6                BEQ  HANDL4
17CC  20  6A  1F            JSR  GETKEY      return key # in A
17CF  C9  14                CMP  #14         is it PC key?
17D1  F0  CE                BEQ  BSTEP       if so, decrement address
17D3  C9  12                CMP  #12         is it + key?
17D5  F0  E1                BEQ  STEP        if so, increment address
17D7  C9  10                CMP  #10         is it AD, DA, or GO key?
17D9  10  D0                BPL  HANDL1      if so, reset toggle
17DB  A4  F9                LDY  INH         shift key 0-F
17DD  8C  E9  17            STY  SAVX        into
17E0  20  0F  1A            JSR  PACKT1      data display
17E3  91  FA                STA  (POINTL),Y
17E5  50  C8                BVC  HANDL2      relative jump
```

* Program starting address, to be manually inserted in NMI vector, 17FA (NMIL) and 17FB (NMIH).

NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW • NEW

Comprehensive 650X Assembler/Text Editor

The Comprehensive resident assembler/editor is a complete system for entering , storing, editing and assembling programs for 650X - based processing systems. Although designed for use with the KIM system, the editor/assembler can be used on any 650X system such as TIM, OSI, Apple, Baby!, etc...

TEXT EDITOR
A program for creating, editing and saving line-numbered text files stored in random-access memory.

Functions supported are:
- Enter new text
- Delete text
- Find designated string in text
- Resequence line numbers
- List specified block of text
- Load text from paper tape or audio cassette
- Dump text file to paper tape or audio cassette
- Transfer control to assembler
- Return to KIM monitor
- Clear text area

Features:
- Line-number orientation for ease of use
- Compatible with any 6500-type system
- Any command preceded with an 'X' is passed to a user-specifiable routine. You can extend the editor to fit your needs.
- Simple interface to paper tape or audio cassette files
- User specified location of text in memory. No restriction on location of text files. Multiple text files may be in memory simultaneously.
- Length of text files limited only be available memory
- Text files are completely relocatable in memory
- Over fifty pages of documentation is provided

RESIDENT ASSEMBLER
A single-pass assembler which accepts the entire 650X instruction set using the standard MOS Technology notation. Source code may be paper tape or memory resident. Object code is always written to memory.

Features:
- Single pass provides source listing, object code, and error messages.
- User may specify input and output routines or use TTY default.
- User-defined symbol table and source location

The complete system occupies 6K and is available on KIM cassette or KIM/TIM paper tape. Specify starting address of $2000 or $E000. Complete source listing is included.

Price: $60.00 - New Jersey residents please include tax.

Order from: ARESCO, 314 Second Ave., Haddon Hts., N.J. 08035.

*****ADVERTISEMENT*****

HERE'S A KLUGE HARP from Ronald Kushnier, 3108 Addison Ct., Cornwell Hts., PA 19020

4

ADDRESS	MACHINE CODE	LABELS	MNEMONICS	COMMENTS
0200	A9 01		LDA #01	Set PA0
0202	8d 01 17		STA PADD	as output
0205	A0 00		LDY #00	
0207	B9 01 00	(5)	LDA 01,Y	
020A	85 EE		STA EE	
020C	A9 52	(3)	LDA #52	
020E	8d 07 17		STA 1707	
0211	EE 00 17	(2)	INC PAD	
0214	BE 00 00		LDX 00,Y	
0217	CA	(1)	DEX	
0218	d0 FD		BNE (1)	
021A	Ad 07 17		LDA 1707	
021d	29 80		AND 80	
021F	C9 80		CMP 80	
0221	d0 EE		BNE (2)	
0223	C6 EE		DEC EE	
0225	d0 E5		BNE (3)	
0227	A9 80		LDA #80	
0229	8d 07 17		STA 1707	
022C	2C 07 17	(4)	BIT 1707	
022F	10 FB		BPL (4)	
0231	C8		INY	
0232	C8		INY	
0233	B9 00 00		LDA 00,Y	
0236	C9 00		CMP #00	
0238	d0 Cd		BNE (5)	
023A	4C 05 02		JMP 0205	

Ronald Kushnier
3108 Addison Ct.
Cornwell Hts. PA. 19020

The following Kluge Harp uses the driver shown in the Kim User Manual, Page 57.
Notes are stored in even locations starting at 0000.
Time duration of the notes are located in odd locations starting at 0001.
The end of song is sensed by a #00 in the note data.

The notes are as follows:

NOTE	DATA
Low B	EC
C	d8
D	C3
E	B2
F	A5
G	94
A	82
B	74
C	70
High D	6E

Time duration for each note is as follows:

	Time	Data
Whole note	1.6 Sec.	10
Half note	0.8 Sec.	08
Quarter note	0.4 Sec.	04
Eighth note	0.2 Sec.	02
Sixteen note	0.1 Sec.	01

For other times, interpolate!

ROW THE BOAT ASHORE

```
0000   d8 02 B2 02 94 03 B2 01 94 02 82 02 94 04 B2 02
0010   94 02 82 08 94 04 B2 02 94 02 94 03 B2 01 A5 02
0020   B2 02 C3 04 d8 02 C3 02 B2 06 C3 04 d8 04 d8 02
0030   B2 02 94 03 B2 01 94 02 82 02 94 04 B2 02 94 02
0040   82 08 94 04 B2 02 94 02 94 03 B2 01 A5 02 B2 02
0050   C3 04 d8 02 C3 02 B2 04 C3 04 d8 04 00
```

LIGHTLY ROW

```
0000   94 04 B2 04 B2 08 A5 04 C3 04 C3 08 d8 04 C3 04
0010   B2 04 A5 04 94 04 94 04 94 08 94 04 B2 04 B3 03
0020   A5 04 C3 04 C3 04 d8 04 C3 04 94 04 94 04 B3 10
0030   C3 04 C3 04 C3 04 C3 04 C3 04 B2 04 A5 08 B2 04
0040   B2 04 B2 04 B2 04 B2 04 A5 04 94 08 94 04 B2 04
0050   B2 08 A5 04 C3 04 C3 08 d8 04 d8 04 94 04 94 04
0060   B2 10 00
```

Here's another good number from Stan Ockers which will prove useful when giving KIM demonstrations at your next club meeting or maybe for your family when they begin to wonder what happened to the household budget.

***** GET PROGRAMS *****

by Stan Ockers

THIS PROGRAM ALLOWS YOU TO CREATE A LIBRARY OF OTHER PROGRAMS WHICH ARE THEN LOADED INTO EXTRA MEMORY AREA. WHEN RUN, THE PROGRAM WILL LIST THE AVAILABLE PROGRAMS ON A TVT ALONG WITH ASSOCIATED NUMBERS (0-F). PRESSING THE ASSOCIATED NUMBER ON THE KIM KEYPAD WILL CAUSE THE PROGRAM TO BE TRANSFERRED INTO THE REGULAR KIM MEMORY WHERE IT WILL BE STARTED. THE STARTING LOCATION OF THIS PROGRAM IS LOADED INTO 17FA AND 17FB SO THAT PRESSING THE STOP BUTTON ON THE KIM WILL DISPLAY THE CHOICES AGAIN.

```
2000        A0 00       LDY #$00        INIT. INDEX
  02    B9 D0 20 MORE   LDA 2030,Y      GET CHARACTER
  05        C9 FF       CMP #$FF        FINISHED?
  07        F0 0A       BEQ WAIT        YES
  09        84 E6       STY 00E6        SAVE INDEX
  0B    20 A0 1E        JSR OUTCH       OUTPUT CHAR.
  0E        A4 E6       LDY 00E6        RETURN INDEX
  10           C8       INY             INC. INDEX
  11        D0 EF       BNE MORE        UNCOND. JUMP
  13    20 1F 1F WAIT   JSR SCANDS      WAIT FOR KEY
  16        F0 FB       BEQ WAIT        NO KEY DOWN
  18    20 6A 1F        JSR GETKEY      WHAT KEY
  1B        C9 15       CMP #$15        VALID KEY?
  1D        10 F4       BPL WAIT        NO
  1F           A8       TAY             USE AS INDEX
  20        A2 07       LDX #$07        8 VALUES FROM TABLE
  22    B9 50 20 TABL   LDA 2050,Y      GET POINTER
  25        95 E7       STA 00E7,X      STORE IT
  27           98       TYA             UPDATE INDEX
  28           18       CLC
  29        69 10       ADC #$10
  2B           A8       TAY
  2C           CA       DEX             MORE VALUES?
  2D        10 F3       BPL TABL        YES
  2F        A6 EA       LDX 00EA        PAGES TO TRANSFER
  31        F0 14       BEQ REMA        LESS THAN ONE PAGE
  33        A0 FF       LDY #$FF        INDEX FOR NEXT PAGE
  35        B1 EB MORE  LDA (00EB),Y    GET BYTE
  37        91 ED       STA (00ED),Y    MOVE IT
  39           88       DEY
  3A        C0 FF       CPY #$FF        LAST BYTE?
  3C        D0 F7       BNE MORE        NO
  3E        E6 EC       INC 00EC        UPDATE PAGES
  40        E6 EE       INC 00EE
  42           CA       DEX             ANY MORE PAGES?
  43        30 06       BMI PGRM        NO
  45        D0 EE       BNE MORE        YES
  47        A4 E9 REMA  LDY 00E9        PART OF A PAGE
  49        D0 EA       BNE MORE        UNCOND. JUMP
  4B    6C E7 00        JMP (00E7)      INDIRECT JUMP
```

VALUES MUST BE PLACED IN THE FOLLOWING TABLE TO INDICATE WHERE LIBRARY PROGRAMS START, WHERE THEY ARE LOCATED AND HOW MANY BYTES THEY CONTAIN. THE VALUES FOR PROGRAM #0 ARE STORED IN 2050,2060,2070 ETC. THE VALUES FOR PROGRAM #1 ARE STORED IN 2051,2061 ETC. UP TO 16 PROGRAMS CAN BE REFERENCED.

```
2050-205F  TO LOCATION, HIGH (PAGE IN KIM-1 WHERE PROGRAM IS TO
                              BE LOADED)
2060-206F  TO LOCATION, LOW (LOW VALUE TO GO WITH ABOVE)
2070-207F  FROM LOCATION, HIGH (PAGE WHERE PROGRAM IS FROM)
2080-208F  FROM LOCATION, LOW
2090-209F  # OF PAGES TO TRANSFER
20A0-20AF   # BYTES IN ADDITION TO THAT NUMBER OF PAGES
20B0-20BF  STARTING LOCATION OF PROGRAM, HIGH
20C0-20CF  STARTING LOCATION OF PROGRAM, LOW
```

LOCATION 20D0 AND FOLLOWING CONTAINS THE TEXT OF YOUR LIBRARY LISTING FOR PRINTOUT ON A TVT. ASCII EQUIVALENTS ARE USED. BE SURE TO INCLUDE THE APPROPRIATE CONTROL CHARACTERS AND END YOUR TEXT WITH A "FF".

THE PROGRAM CAN BE USED WITHOUT A TVT, JUST KEEP A LIST OF THE PROGRAMS AND ASSOCIATED NUMBERS. PUT A "FF" IN 20D0 OR JUST START THE PROGRAM AT 2013 (REMEMBER THE VECTOR AT 17FA AND 17FB).

IF YOU HAVE FEWER THAN 16 PROGRAMS, 201C SHOULD CONTAIN THE NUMBER OF PROGRAMS YOU HAVE PLUS ONE (IN HEX).

Now we can learn what codes our keyboards really put out and become more familiar with KIM monitor routines in the process....Eric

PAPER WASTERS — from... Charles R. Carpenter, 2228 Montclair Place, Carrollton, TEXAS 75006

For new KIM-1 programmers like myself, here are a couple of routines to help learn the machine and uses of some of the monitor sub-routines (as suggested by Eric in the complementary issue). The first routine will get a character from the TTY keyboard, display it and print the hex value for the character. I found this little routine useful for learning all the codes that are generated by my keyboard and associated electronics. Also, I learned some things about what the machine will allow in trying to use the various sub-routines together. The second routine will print the same information, but only if the hex value of the character is loaded into the accumulator first. Any other valid data could be used for the LDA value. By selectively using spaces (1E9E), carriage returns (1E2F) and characters (1EA0) a matrix of characters in rows and columns can be generated (poor man's graphics). Let the Users Notes know if you come up with any other combinations. Have fun.

Routine No. 1 Loop

```
0000    20 5A 1E    JSR     1E5A        Load accum. & print char. from TTY
0003    85 16       STA     16          Store char. from A in memory
0005    20 9E 1E    JSR     1E9E        Print a space
0008    A5 16       LDA     16          Load accumulator with memory
000A    20 3B 1E    JSR     1E3B        Print hex code for char. in A
000D    20 9E 1E    JSR     1E9E        Print a space
0010    20 9E 1E    JSR     1E9E        Print a space
0013    4C 00 00    JMP     0000        Return to start for next char.
0016                        Scratch Pad (Relocate as needed)
```

Routine No. 2 Loop

```
0000    20 2F 1E    JSR     1E2F        Line return
0003    A9 41       LDA     #$41        Load accumulator with char. in hex
0005    20 A0 1E    JSR     1EA0        Print char. in accum. (41 hex A)
0008    20 9E 1E    JSR     1E9E        Print a space
0008    A9 41       LDA     #$41        Load char. again
000D    20 3B 1E    JSR     1E3B        Print hex code for char.
0010    20 9E 1E    JSR     1E9E        Print a space
0013    20 9E 1E    JSR     1E9E        Print a space
0016    4C 00 00    JMP     0000        Return to start of program
```

NOTE: Start and end at 0000 this routine prints a column - start and end at 0003 prints a page until it is reset.

Judging from my recent mail, a good number of you are planning to add Lancasters TVT-6 to your system. If so, then you'll be interested in what Jim Butterfield has to say on the subject. The TVT-6 has got to be one of the neatest developments to come down the pike yet and it's got me to wondering what Lancaster's next trick will be...Eric

Notes on Don Lancaster's KILOBAUD article, "A TVT for your KIM"

by Jim Butterfield

A great article, with good material in it. I'll try to explain in more detail how it works.

First, a word of caution. You'll have to "chop up" your KIM a bit to implement this - the project involves cutting a piece of KIM's printed circuit foil, plus wiring in a whole bunch of new wires. And while the changes don't affect KIM's operation, you have to recognize that memory expansion becomes a different ball game. Don uses the addresses from 2000 to EFFF, and that means that you can't just add on extra memory in those areas.

Much of the operation relies on Don's upstream tap. To get an idea of this, check your KIM user manual, page 27 (Fig. 3.4). Data comes out of the RAM memory (U5 to U12) from pin 12, and goes straight to a gate (U13 and U14).

Originally, this gate was there to block the data out if you were writing to the RAM. Now, when the TVT is enabled with an address from 2000 to EFFF, the data is blocked anyway. Instead, pin 12 feeds directly to the display character generator. And the main data bus, instead of reading memory, gets a dummy code A0 (Load Y) fed to it from ROM (IC2).

What it means is this: when theprocessor branches to 2000, it thinks it's reading LDY #$A0 from memory. But page zero memory is feeding completely different data straight to the display! The LDY instruction that the processor sees executes fast, in two microseconds, so that the address bus goes clipping right along at 1 microsecond speed. As the address bus steps, it simultaneously delivers page zero characters to the display, and the ROM code A0 to the processor.

When we reach the end of a lone, the SCAN ROM finally delivers code 60 instead of A0, and the microprocessor returns to normal memory and normal activity. Of course, to keep the display going, we will need to JSR back to this program very quickly to catch the next scan line.

In fact, you won't start writing live data to the screen until you give the command JSR 2200. This must be followed with JSR 3200, JSR 4200, JSR 5200, and so on until JSR D200 (each instruction sends a different part of the characters, starting at the top); now you've sent a complete line. Send a blank scan line to separate the next line of characters (JSR 2000), and now you can start this line with JSR 2220, then 3220, etc.

In case you didn't catch it, the sequence starting with JSR 2200 displays memory 0200 to 021F; when you start JSR 2220, that will display from 0220, and so on. The sequence continues until you get to the JSR 23E0 sequence, at which time you've displayed the full memory of 16 lines.

Fast Tape: a status report — Jim Butterfield, Toronto

Name Change: to avoid confusion with a cassette tape brand name, let's call the high-speed tape (formerly supertape) by a new name: Hypertape. I'll use the term Hypertape from here on.

Most reports on Hypertape (formerly supertape) are that it's 100% reliable. Difficulties are uncommon, and are usually caused by:

--Failure to write good Hypertape: dirty tape head, worn tape head, poor electronics especially the bias oscillator. I often get CB interference on my tapes; oddly, they still work OK.

--Failure to read Hypertape on the same machine as recorded: unregulated 12V supply to KIM, low volume levels.

--Failure to read Hypertape on a different machine: almost always discrepancies in head alignment between the two machines.

This last item - incompatibility between the read and write machines - can usually be overcome by dropping to half Hypertape speed (Speedtape). This is still three times faster than normal tape. I suggest you use it when mailing a tape to a distant friend. Eric Rehnke realigns his tape head to match each Hypertape he receives, which also works OK; but not everybody is prepared (or able) to do this.

New Directions

Hypertape is plenty fast for me with my 1K system, but others are working on further speedups, which could be useful for large memories.

Julien Dubé, who had a lot to do with the birth of Hypertape, is making considerable progress with a new idea of mine. The idea is this: if you strap pin E-X to A-T, signals coming in from cassette will be seen by KIM as teletype input. In fact, you can print them on a teletype if you have one, because they feed back to pin A-U; you'd need the right speed, of course. (A paper tape simulator?!) Writing the signals to cassette is a small project, since you must put tones, rather than DC signals, onto the tape.

Since we're not tied to mechanical teletype speed, these signals can be speeded up to a fantastic rate, say 2400 baud. At the moment, Julien is using standard paper tape KIM format, and using the ROM program, starting at 1CE7 to successfully load memory from cassette. Eventually, a separate load program may be written. Potential: about 4 times faster than Hypertape. Julien's new address, by the way, is 3174 Rue Douai, St. Foy, Quebec, Canada.

Hal Gordon (Oakland, CA) is working on another approach. Instead of writing frequencies to tape, he's writing the bits directly! To read this back, the PLL (phase lock loop) input of the KIM is bypassed and the arriving bits go directly to the processor. A hardware interface is required, of course. The speed potential of such an approach is fantastic; and Hal reports considerable success in his early test shots. He has plans to build in extensive error routines, and is thinking in terms of a Super-Loader program with many features not in the KIM loader.

I'm glad to see some of you have put on your academic hats to help us out with our problems... Eric

CASSETTE TAPE INTERFACE NOTES — by Dwight D. Egbert

I have noticed several comments in the Users Notes about cassette read/write problems which reminded me of a problem I encountered while building a 1200 Baud KC standard cassette interface for my 8080 system. I used the same input scheme with back to back diodes for limiting (CR1 and CR2, Figure 3.8, p. 31, KIM-1 Users Manual). My problem was caused by some asymmetry in the diode forward conducting properties which caused a voltage offset and a highly skewed signal which confused the frequency discrimination circuit (which was not, however, a phase lock loop like KIM-1). The fix for this was to put another diode (1N3600) in parallel with one or the other of the original diodes (1N914) in the direction to correct the offset. With this in mind I decided to take a look at my KIM-1 cassette circuit (which has always worked fine).

I did not find any offset problems (probably because of R14 and R15), but I did make several other observations that might explain some tape read problems. First, the input signal voltage is reduced by a factor of ten across R8 (10Kohm). This means that unless your tape recorder puts out more than +/- 6 volts (12 volts peak to peak AC) the diodes will not saturate and do any clipping of the signal at all. In this case you are inputting an unaltered signal across R14 and R15 (which is perfectly alright). In order to test the circuit sensitivity at these low levels I reduced the output level of my recorder until read errors started to occur with a known good tape. This occured (for my KIM-1) at approximately 0.25 v. p-p which produced about 0.025 v. p-p across pins 2 and 3 on U27 (LM565).

continued...

hile leaving the tape level constant I placed another 10Kohm resistor in parallel ith R8 which raised the U27 input to 0.05 v. p-p. The tape read properly at this oint as well as at higher input levels with the parallel resistor in place. his resistor changes the divide by ten to divide by five and is equivalent to eplacing R8 with a 5Kohm resistor (1/10+1/10=1/5). If you are using a true Aux. utput from your tape recorder you are probably only getting less than 1.0 v. p-p hich puts you down near the questionable levels. In this case reducing R8 ould help reduce read errors. Even with R8=5Kohm the diodes will prevent cessive signal levels from reaching the input of U27 and any input that will mage the circuit at 5Kohms will also probably do damage at 10Kohm. Alternativ-y, if your Aux. jack is really a speaker output as is the case with most rtable recorders you will have plenty of voltage. My recorder is like this $79 Panasonic) and produces super results at an output level around 5 v. p-p. is gives about 0.5 v. p-p at U27 which is around 10 times the threshold level.

Another thing I noticed was that for the particular tape I was reading the wer frequency signal was about 25% larger amplitude than the higher frequency. this difference in amplitude gets too large the reliability of your frequency scrimination can be impaired. You can reduce this problem by setting the tape corder tone control near maximum treble.

If you are having serious tape problems one of these three possible fixes ght help 1) add a diode, 2) reduce R8, or 3) use more treble. However, I have und through experience that it is far more likely that your tape read problems e caused by either tape drop-outs or dirt. To alleviate these two plagues I ean my recorder often, particularly before recording, and use only two brands tape in C30 or C45 short cassettes. Radio Shack Supertape and Memorex MRX2 e both good tapes, even at 1200 Baud. Scotch High Density is bad. Also, I ways make at least two copies of all files. Considering the usual manual covery time if a file is lost, tape is cheap!

Dwight D. Egbert
302 W. 109, #4
NYC, NY 10025

ADDING A HIGH-SPEED PAPER TAPE READER TO KIM is fairly simple. Most any low cost optical e reader equiped with parallel data output and "hand shaking" capability can be used.

With the exception of the GETCH subroutine ($1E5A), the KIM paper tape program ($1CE7-1D40) the GETBYT sub ($1F9D) can be copied out into ram somewhere. The subroutines, PACK ($1FAC) CHK ($1F91) can be left in rom. All references to GETCH and GETBYT must be changed to lect the new addresses of the modified routines. The new GETCH should loop around until the e reader sends a data ready strobe, read the character into the accumulator from PAD, send ata received strobe back to the reader, strip off the parity position and return to the main gram. This routine must also preserve the "X" register and return with "Y"=$FF to simulate original GETCH routine. PB7 can be used as the "data ready" strobe input from the tape der and PB0 as the "data receive" strobe output from KIM. PBDD ($1703) should be initialized $01 at the start of the main reader program.

Here's an idea for the new GETCH

```
(new)  GETCH    BIT PBD    check for data ready strobe
                BPL GETCH  depends on polarity of strobe
                LDA PAD    get character
                AND #$7F   strip off parity
                INC PBD    send strobe          } no-ops may need to be added
                                                } here if longer strobe pulse
                INC PBD    for data received    } is necessary.
                LDY #$FF   to simulate original GETCH
                RTS        and return
```

Since the "X" register is not modified, it need not be protected.

-the editor-

****

The MICROTERM ACT-1 looks to be a popular, reasonably priced, 64x16 standalone terminal. With Mr. Carpenter's help, getting it on line with KIM should be easy... Eric

KIM-1, ACT-1: THE SCENE

Charles R. Carpenter
3139 Montclair Place
Carrollton, TX 75006

I recently purchased a MICRO-TERM INC. ACT-1 TTY replacement terminal and, after resolving some interfacing problems, I have it running with my KIM-1. Hookup data supplied with the unit is very general and I would like to share my experience with KIM-1 users.

After making all the external connections and one internal change per the users manual, I was unable to get the ACT-1 running. I made a few phone calls to MICRO-TERM but the results were still negative. The people at MICRO-TERM were very cooperative but unfamiliar with the KIM-1. I finally got up enough courage to experiment and the results that worked are as follows:

Internal Connection	Connect To
Serial Output Level	P
Serial Polarity Out(put)	Invert
Serial Polarity Input	Unchanged

Part of the confusion comes from the serial output level marking on my board (ACT-1, 4-77, REVD). It is wrong according to MICRO-TERM. The only other problem was an unsoldered key switch. I could not get one character to print. After soldering the connections, everything was fine.

I have the baud rate set at 1200 and have had no problems using the system at this rate. The screen will fill completely in about 20 seconds. I can display a little more than 256 bytes (one KIM page) for each memory dump. This includes the start address and format characters plus the ending line which uses up some of the space. (My SX70 camera works fine for making a hard copy of the program if I want one.) By setting the interrupt vectors at 17FA-FF to 1C00, I was able to use the ST key to stop the run and examine it at any point. Typing RETURN (after ST) and then G again when ready, started the run at the last address indicated after RETURN was typed. This worked only when the ending address at 17F7-F8 was set at 2000.

MICRO-TERM has done a good job on the ACT-1 and I would recommend this unit to anyone planning to include a serial TVT terminal in their system. I hope that other users get the same enjoyment using the ACT-1 that I have.

Several other members have mentioned problems with "bouncy" keys. Does anyone have a cure for this problem? Iwould sure like to hear about it.... Eric

Gentlemen:

We have several KIM-1 systems that are being used in our Computer Engineering curriculum for hands-on microcomputer experience. The KIM's have performed beautifully except for one problem. We have had trouble with bouncing "9" keys on some of the keyboards. A replacement keyboard (a new one) was installed in place of a bouncy one and then it was discovered that it too was bouncy. Is this a problem that has plagued other KIM-1 users? Also, is there anyone who can provide a satisfactory solution to this problem of bouncy keyboards? A remedy would be greatly appreciated.

Thank you.

David A. Byrd
Computer Engineering
State Technical Institute at Memphis
5983 Macon Cove
Memphis, TN 38134

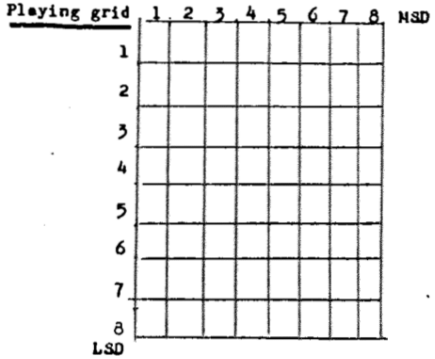
BATTLESHIP GA

by Ronald Kushnier, 3108 Addison Ct., Cornwells Heights, Penna. 19020.

An enemy battleship resides in three adjacent squares of an 8 x 8 matrix. Your mission should you decide to accept, is to try to sink the battleship by three direct hits.

Enter AD 0200 and press Go. The display indicates as follows:

Enter your choice of coordinates: 1, 1 to 8, 8
Press F (Fire)
Continue until you're out of ammo or the ship is sunk
If you run out of ammo the three coordinates of the ship will be displayed
Note: Battleship is placed randomly by KIM and may be positioned horizontally, vertically or diagonally on the grid.



Send a S.A.S.E. for the listing.

PROGRAM OUTLINE

Grid created in page zero
Actual grid looks as follows:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	02	02	02	02	02	02	02	02	02	02	02	02	02	02	02	02
1	02	00	00	00	.	.	.	.	.	02	02	02	02	.	.	.
2	02	00	00	.	.	.	.	.	.	02	02	.	.	.	.	.
3	02	.	.	.	.	.	.	.	.	02	.	.	.	.	.	.
4	02	.	.	.	.	.	.	.	.	02	.	.	.	.	.	.
5	02	.	.	.	.	.	.	.	.	02	.	.	.	.	.	.
6	02	.	.	.	.	.	.	.	.	02	.	.	.	.	.	.
7	02	.	.	.	.	.	.	.	.	02	.	.	.	.	.	.
8	02	.	.	.	.	.	.	.	.	02	.	.	.	.	.	.
9	02	02	02	02	02	02	02	02	02	02	02	02	02	02	02	02

The grid is formed by inserting all 02's (line 0200-020F); then, selectively inserting 0's in the active area (0211-022B).

RAND (022D-023D) generates a random number 0-99. The direction of the ship is selected by looking at the least significant 2 digits and decoding accordingly. (023F-028A) 00←, 01↑, 10↙, 11↘. Also in this section the computer looks for 2's which indicate the perimeter of the grid. If a 2 is found, the computer throws out the random variable and tries another.

The display portion of the program consists of loading the proper initial values in the LEDS and waiting for a two digit keyboard coordinate (028D to 02DA). A mistake can be corrected in a manner similar to the KIM monitor. Coordinates not included in the grid will not be entered.

Upon depression of the F (Fire) button, the program (02DD) checks to see if there was a hit, if so it increments the hit count and decrements the ammo left. If no ht, it just decrements the ammo. Also to preclude the operator from continually firing on the same coordinate and thereby getting the required three hits, after the first hit on a "good" coordinate any additional fire will just decrement the ammo.

If three good hits are attained, the display will read "dead". The ship is sunk (0311).
If "out of ammo" is reached, the coordinates of the ship will appear (0320).

Zero page data storage 00E3 - 00EE.

BATTLESHIP

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0200	A9	02	85	00	A9	00	85	E8	A2	99	A9	02	95	00	CA	D0
0210	F9	A9	11	85	E7	85	E3	A2	07	18	A0	07	A9	00	91	E7
0220	88	10	FB	F8	A5	E7	69	10	85	E7	CA	10	EC	38	A5	EA
0230	65	ED	65	EE	85	E9	A2	04	B5	E9	95	EA	CA	10	F9	29
0240	03	C9	00	F0	41	C9	01	F0	36	C9	02	F0	19	18	A0	02
0250	A6	E9	B5	00	C9	02	F0	B9	A9	01	95	00	8A	69	11	AA
0260	88	10	EF	4C	8D	02	A0	02	A6	E9	B5	00	C9	02	F0	A1
0270	A9	01	95	00	8A	38	E5	E3	AA	88	10	EE	4C	8D	02	A9
0280	10	85	E3	4C	66	02	A9	10	85	E3	4C	66	02	A9	20	85
0290	FA	A9	00	85	F9	85	E4	85	FB	85	E6	D8	20	1F	1F	20
02A0	6A	1F	C9	0F	F0	37	C9	09	10	F1	C9	00	F0	ED	85	E5
02B0	A5	E6	C9	01	F0	16	E6	E6	06	E5	06	E5	06	E5	06	E5
02C0	A5	E5	85	FB	20	FE	1E	D0	FB	4C	9B	02	18	A5	E5	65
02D0	FB	85	FB	C6	E6	20	FE	1E	D0	FB	4C	9B	02	A5	FB	C5
02E0	E4	F0	07	AA	B5	00	C9	01	F0	17	F8	A5	FA	38	E9	01
02F0	F0	2E	85	FA	D8	A5	FB	85	E4	20	FE	1E	D0	FB	4C	9B
0300	02	E6	F9	A5	F9	C9	03	F0	08	20	FE	1E	D0	FB	4C	EA
0310	02	D8	20	1F	1F	A9	DE	85	FB	A9	AD	85	FA	4C	11	03
0320	A0	02	A2	99	B5	00	C9	01	F0	06	CA	D0	F7	4C	38	03
0330	8A	99	F9	00	88	4C	2A	03	20	1F	1F	4C	38	03		

TTY INTERFACE INFO from: John Leslie, 10 Souhegan St., Milford, N.H. 03055

Some people out there would probably like to know that the KIM-1 teletype interface does in fact operate at baud rates substantially over 300. Probably anyone having a terminal capable of higher speeds has noted that it works quite nicely at 600 baud. But they may not have looked into the hardware and software to find out why it doesn't work too well above that.

The trouble you run into at about 1200 baud turns out to be quite simply software which can't be bothered to correct for its own running time. You can get around this quite nicely by changing locations 17F2-3 to an appropriate figure. For 2400 baud I use 19,00 and for 4800 baud (on good days) I use 0A,00. An annoying feature of the software is that it sets 17F3 to FF on RESET, but you can quickly learn to fix that with the on-board keyboard.

The trouble you run into at about 4800 baud is the noise-limiting capacitor C5. This is the .33 mfd. capacitor parallel to the top of the on-board keyboard. Although I have not yet tried reducing it (4800 baud works most days, and is quite livable), I can imagine no reason why it should be that big except for operation at low baud rates.

At about 9600 baud, you would run into the software problem again, in that it ignores its own execution time, and puts out quite incorrect waveforms and reads the bits at seriously wrong times. Using software which corrects for its execution time and reducing C5 to roughly .022 mfd., I would expect you could get very dependable operation at 19,600 baud.

CLEVELAND COMPUTERFEST REPORT

The second annual Cleveland Computerfest was probably twice as good as last year. KIM certainly was well represented this year. I got the opportunity to meet with a number of our group and also had a chance to meet Wayne Green (73 and Kilobaud publisher) and chat with him for a few minutes. He's quite a KIM supporter, you know.

Rick Simpson of ARESCO was there with a 4K FOCAL interpreter and a couple of assemblers. Talk about great documentation - that FOCAL listing reads like a book. It's worth getting a copy just to see how the language is put together. Rick mentioned that an improved 6502 FOCAL will be released shortly featuring such enhancements as; 30% faster execution time; improved string handling capability; and an interrupt handling facility.

By the way, Rick Simpson is now back at MOS Technology so we should begin to see a little more in the way of KIM development. Look for ARESCO at Atlantic City PC '77.

An impressive showing of expanded KIM systems was displayed by the KIM faction of SEMCO (Southeast Michigan Computer Club). They had the Game of Life running on a KIM driven MATROX video display and also had a neat music program going. Their machines were expanded using the 44-pin bus (similiar to KIM-4) and an interface card (about 4" wide) which slipped onto KIM's edge connector. Ribbon connectors were used to interconnect the system.

These fellows wasted little time (seemed like 15 min.) in getting FOCAL up on one of their machines. You'll be hearing more from this group. They've also been putting together some dynamite 6502 software (system level stuff) and are working on developing software standards. (I've been promised an article on this as soon as they wrap it up!) A member of the group, Rene Vega, will be introducing a KIM expansion system based on the 44 pin bus and the 4.5" by 6" card size shortly. More word on this when it's released.

I met with Peter Jennings (MICRO-WARE LTD) who was putting the finishing touches on his 4K Assembler, Editor, Disassembler package. (See pg. 1). Judging by the way he crammed a chess game into 1K of memory, this 4K package should be something. Peter will also be at Atlantic City.

Riverside Electronics showed off their MVM-1024 video display board and the KEM (KIM to S-100 bus adapter). I was especially impressed with the versatility of their video board. Rather than taking up a 1K slot in memory, the board decodes as 3 I/O ports. Two of these ports are for cursor control and enable you to read or write the X,Y cordinate of the cursor directly. This makes any location on the 64x16 display immediately accessible for a read or a write command.

Riverside also has a KIM-1/6502 display driver program available.

One fellow gave a seminar on computer controlled model railroad using the KIM (what else). Very impressive.

All in all, we had a great time in Cleveland and are now looking forward to the PC '77 in Atlantic City in August. Hope to see you there.

Eric Rehnke

MINI-L LORAN-C FRONT-END FOR µP EXPERIMENTER & TIME-INTERVAL MEASUREMENTS
A few sets of two circuit boards and a 25 page users manual for the assembly of a Loran-C 100 kHz RF front-end system called Mini-L are available. This is a pulse format navigation system with shorter range but more precise than Omega (see my articles in BYTE, Feb,Mar, Apr, 1977 on related Mini-O). Mini-L is designed to provide a synchronized 10 µsecond interrupt request for each Loran-C pulse envelope, which user must manipulate with his own software or hardware to measure time intervals. No parts are supplied, only the basic circuit boards and suggestions on interfacing. Cost $21 shipped by 1st class mail, send check or money order to R. W. Burhans, 161 Grosvenor St., Athens, Ohio 45701, NO COD. Software will be available in a few months and another publication is anticipated about the Mini-L system in the future. In the meantime, experimenters skilled in the art of receiver fabrication and use of µP systems, can study precision time-frequency measuring problems with Mini-L at a cost about 1/100th of the lowest cost commercial Loran-C system available.

Book Review:

CMOS COOKBOOK by Don Lancaster

publication #21398 / $9.95
Howard W. Sams & Co. Inc.

Lancaster covers a surprising amount of ground within the pages of his latest "cookbook".

Much like his previous books, he starts off with an explanation of the particular logic family and includes a course in logic fundamentals starting off with one input gate and moves thru flip-flops, counters, multivibrators, etc. Lancaster then moves out of the purely digital realm by introducing such exotic things as CMOS op-amps and phase-locked loops. Basic theory and design rules are presented to help you get started with these neat devices.

I like the way he blends practical examples into the discussion. It tends to keep up your enthusiam even when the theory seems a little difficult to comprehend at first.

Lancasters' cleverness will jump out at you when you see how he implements a "tracking" active filter section by use of a CMOS analog switch.

Plenty of info is included to assist you in interfacing various things to your micro. It looks like CMOS will prove particularly useful in this area with its low power, design simplicity, and good noise performance. A whole gang of CMOS LSI parts such as touch tone generators, top-octave music generators, DVM chips, frewuency counters, modems, etc. etc. are available to make life easier for you and alot of these chips are included in the CMOS mini catalog chapter of the book.

The "system" level design section includes schematics and theory for such things as: an all CMOS TV typewriter; a basic music synthesizer, an electronic stop watch etc. etc...

Every state-of-the-art hobbyist (or engineer) should have the CMOS Cookbook on his bench. It's the kind of book that never seems to get put back on the shelf.

Eric Rehnke

SMALL MICRO CONTROLLER BOARD USING 6505

I would like to anounce the developement of a small controller board using a 6505 CPU. The 6505 is the same as the 6502 with fewer address lines and in a 28 pin package. I have designed the KIM into a number of projects and then wished that I had a dedicated controller to perform that task so that the kim would be free for other things. This led me to design a small board with the same micro so that the software which I had could be used directly.

The board contains a 6505CPU, 2- 1702's, one page of RAM (2112), and 12 input lines and 6 output lines. It also has provisions for an interrupt latch and reset. On-board power supply too.
We are presently in the process of design a micro-controlled repeater/autopatch in the Tulsa area using one of these boards. Also several of the local hams have taken them and are using them with ASCII keyboards for RTTY. (Have a program which handles the ASCII to BAUDOT, fifo, memory, and other things.)

The boards are double sided, Plated-through holes and about 6"X4". I have been having them built by the batch, as the need appeared. If there is enough interest, I will be glad to have some more of them built up. I will sell them for 15.00 ea. including schematic and assembly instructions. (less Parts).

CONTACT: Dan Bates Rt 7 Box 310 Claremore, Okla. 74017.

KLUGE HARP I ND EDITOR CORRECTION

from Ted Beach, 5112 Williamsburg Blvd., Arlington, VA. 22207

It was very gratifying to see my two utility programs in the expanded KUN#3, Eric, however, I must take exception to your remark that the PATCH program "... will not cross page boundries..." Indeed it will. That is the reason for the two instructions at 17A7 and 17A9. Regarding the matter of being able to move data up only, this was strictly intentional since most programs are straight-line and extra space is available only at higher addresses.

Your readers might also be interested in a "fix" I made to the KIM-1 KLUGE HARP of Robert G. Lloyd (KUN#2, page 7). As written, the instruction at 032F is wrong, and the program will "run on" until it sees data the same as is at location 0030 (or 0063 for DAISY). The revised program is shown, starting at 032F. Now all you have to do is write FF after the last note of the music field (in page zero) to halt the program - no need to count bytes or change location 0330. Also, the program will halt at 0300 so you can do it again by pressing GO. Incidentally, I use a simple transistor buffer (as in the KIM manual) to drive the loudspeaker.

Also, with regards to the KIM-1 KLUGE HARP, I find that, even with my tin ear, the note values Bob Lloyd gave just don't quite make it for me. I am including my revised listing (whole notes only) which sounds a bit better to me.

KIM-1 KLUGE HARP "FIX"

```
032F C9 FF          CMP #FF
0331 D0 CF          BNE LOOP2
0333 A9 00          LDA #00
0335 8D 1E 03       STA NOTE
0338 85 FA          STA POINTL
033A A9 02          LDA #02
033C 8D 16 03       STA 0316
033F A9 03          LDA #03
0341 85 FB          STA POINTH
0343 4C 4F 1C       JMP START
```

OCTAVE	C	D	E	F	G	A	B
LOW	B8	A2	90	87	79	64	5E
MID	59	4E	45	40	39	33	2B
HIGH	28	22	1D	1C	18	-	-

PATCHES FOR MCS6502 RESIDENT TWO-PASS ASSEMBLER/TEXT EDITOR

Jodie S. Hobson, 1104 N. Overhill Ct., Wilmington, De. 19810

I have discovered a bug in the KIM version of the resident two-pass assembler/text editor currently being marketed by MICRO SOFTWARE SPECIALISTS, INC. The bug prevents the operation of the break key during the listing of a program as described in the documentation, and can cause the assembler/text editor to die when you try to list your program or whenever it feels compelled to print an error message. The reason appears to be that the code to sense the break key was not translated from the original TIM code. The correct KIM code is:

```
47BC  48              PHA
47BD  AD  40  17      LDA SAD
47C0  2A              ROL
47C1  90  04          BCC BREAK
47C3  68              PLA
47C4  4C  E8  47      JMP WRT
47C7  AD  40  17  BREAK LDA SAD
47CA  2A              ROL
47CB  90  FA          BCC BREAK
47CD  4C  E0  46      JMP $46E0
```

Also, for those people who want to change the I/O linkages, there is an undocumented call to the write character routine at address 477D.

More on the MICRO-SOFTWARE Assembler/Editor

From; Robert L. Kurtz, #4 Santa Bella Rd., Rolling Hills, Ca. 90274.

.....Incidentally, we got the assembler/text editor tape from MICRO-SOFTWARE SPECIALISTS (Commerce, Tx.) and it works fine --- if you clear up a little "glitch". In the 4000-4830 tape - location 4409 is 27 but should be 47! Drove us up the wall for 2 evenings".

MICROCOMPUTER PROGRAMMING COURSE

Chaffey Community College at Alta Loma, Calif., has scheduled a full-quarter course in cimrocomputer programming featuring the KIM-1 that will be offered in two 12-week class sections.

Both classes will be from 7 to 10 p.m., one on Tuesdays starting Sept. 13 and the other on Wednesdays beginning Sept. 14.

The only prerequisite for the three-unit course is that a student be a high school graduate or 18 years of age. There is no tuition fee entailed for state residents.

Further information may be obtained by writing to Donald J. Ketchum data processing professor, Chaffey Community College, Alta Loma, Ca. 91701.

The BAY AREA TVT kits are again available from: BYTE SHOP #2, 3400 W. Elcamino Real, Sante Clara, Ca. 95051 (408) 246-4813.

Basically, its a 32x16 scrolling display that includes a parallel interface and manual cursor control all on a single board. Your editor is presently using this TVT with the SAB-1, serial interface (see Kilobaud #1, pg. 114) as a stand alone TTY type terminal with KIM. It works great! The SAB-1 board is no longer available (too bad).

You should be experienced in digital construction techniques, however, before attempting to build any device of this complexity because the documentation is not up to Heathkit standards. But then not many kits are! My TVT worked immediately after I turned a chip around which I installed backwards. I would recommend sockets for all chips.

This TVT board may be converted to 64 characters without too much trouble.

Price for the complete kit is $140.00 or $20.00 for just the board. (Add $2.00 for postage). Shipping is within 30 days of receipt of order and Mastercharge and Bankamericard may be phone in.

FOR SALE:KIM-2 4K RAM board.New condition with all packing and documention. Owner needs larger unit. $140.00. Contact; J.C. Williams, 35 Greenbrook Dr., Cranbury, N.J. 08512 (609) 448-7782

FOR SALE: KIM-1 microcomputer board, KIM-4 motherboard, and power supply. $300.00 takes it all. Contact: Louis Shapiro, 2429 Surf Dr., Bellmore, N.Y. 11710

PAGE ONE PROGRAMMING PROBLEMS AND A SOLUTION

from: Timothy Bennett, 309 Mary St., Westerville, Ohio 43081

Programming in Page 1

SYMPTOM- The upper limits of my program allow for only 30 Bytes of stack. Various program parameters can be changed through routines which are accessed by stopping and addressing via the KIM-1 keyboard. After accessing routines several times in this manner the stack would start overwriting my program. The effect would be accumulative each time the program was interrupted.

CAUSE- I was interrupting my program by depressing the ST button. If my program happened to be in some level of subroutines, then the stack pointer would not be reset to FF. When the main program was re-started the stack pointer would not be re-initialized.

SOLUTION- If you have programs in page 1 that must be manually interrupted, then use the RS button. This will initialize the stack pointer to FF. Then be carefull not to re-enter your program in a subroutine. Use the ST button only for debugging, and then be aware of its effect on the stack pointer.

SPEED CONTROL OF KIM-1 TTY PORT

If you are having problems trying to use a high speed terminal with the KIM TTY port the following information might be useful. To start with, the Reset/Rubout sequence activates a subroutine called DETCPS (1C2AH to 1C4EH) which determines two constants CNTL30 (17F2H) and CNTH30 (17F3H). These are used to time the serial TTY port via subroutines DELAY (1ED4H to 1EEAH) and DEHALF (1EEBH to 1EFDH). DELAY and DEHALF are called by GETCH (1E5AH to 1E87H) which inputs one character, and OUTCH (1EA0H to 1ED3H) which outputs one character.

CNTL30 and CNTH30 are the whole key to trimming up the TTY port speed. You can change the Baud without going through the Reset/Rubout sequence simply by changing one or both of these constants. The following list gives the values of CNTL30 and CNTH30 for several speeds as determined by my particular KIM-1 Reset/Rubout sequence.

BAUD	110	150	300	600	1200	1800	2400	4800	9600
CNTL30	85H	D8H	EBH	74H	38H	24H	1AH	06H	03H
CNTH30	02H	01H	00H	00H	00H	00H	00H	00H	00H

Now for the interesting part. At lower speeds (110 through 1800 Baud) everything works fine with these values. But, at higher speeds problems arise. Repeated Reset/Rubout sequences kept producing CNTL30 = 1AH for 2400 Baud. This value did work for most functions like examine and fill memory, however, for tape dump (Q command) intermittent characters were lost. This problem was eliminated by substituting either 18H or 19H in CNTL30. All functions worked perfectly for either value.

At 4800 Baud I was unable to use the terminal at all. It acted more like it wanted to work at CNTL30 = 07H rather than at 06H, but no value between 05H and 0AH would make it work. At 9600 Baud the slow functions like examine and fill would work, but tape dump (Q) resulted in severe loss of characters. The problem at these speeds is that the quantization level between allowed speeds (i.e. integer values of CNTL30) is so large that unless you are lucky you will not hit close enough to your terminal speed. Since CNTL30 is used in a software timing loop to count instruction sequences (thus, machine cycles) the high Bauds can be fine tuned with the system clock. You should be able to use the technique described by R. W. Burhans (page 10, issue #5, May 77, KIM-1/6502 Users Notes) to perform this fine tuning.

For casual use, 2400 Baud (or maybe even 110 Baud) is satisfactory with no hardware modifications. However, if you have a dedicated high speed terminal you may find it worthwhile to fine tune either the KIM-1 clock or the terminal clock. This will allow you to utilize the KIM-1 software (like GETCH and OUTCH) at high speeds. After you become accustomed to it, 9600 Baud is nice, 2400 ok, 1200 a bit of a drag, and anything less unbearable. Finally, since CNTL30 = 03H (greater than zero) for 9600 Baud you should be able to fine tune for 19,200 Baud, and maybe (but probably not) for 38,400 Baud.

Dwight D. Egbert
302 W. 109, #4
NYC, NY 10025

A LOW COST GRAPHICS POSSIBILITY

If you're looking for a low cost graphics interface for KIM, then check out the article in Popular Electronics (July 1977). Page 41 describes the RCA CDP1861 video chip and shows how to hook it up to the Cosmac "ELF" microcomputer board. It should also be adaptable to KIM with a little thought.

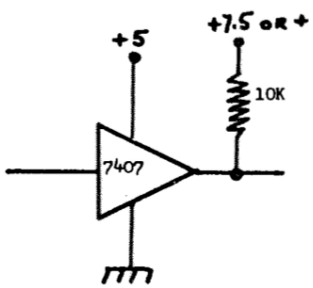
The CDP1861 issues an interrupt signal at a 60 Hz. rate for display refresh and, according to the article, can display up to 1024 bytes of memory (which works out to a 128x64 bit resolution).

The chip could be set up with its own memory so as to ease up the CPU refresh rate and could send out a signal when it's busy so the CPU could wait for display update. Several interesting possibilities exist with this chip, so who'll be the first to get one hooked to KIM???

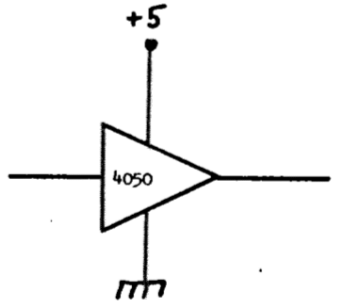
.....Eric

CALCULATOR CHIP SPEED INCREASE by the editor

Hey! Wanna speed up the calculator interface presented in issue #4 ? Simple. Just add the following TTL to MOS level shifters to the four inputs to the calc. chip, and the MOS to TTL level shifters to the nine outputs, then raise the voltage on the calc. chip to about +7.5 or 8 volts. Thats all there is to it! You will notice about a 30% increase in calculating speed.



TTL to MOS level shifter



MOS to TTL level shifter

Speaking of calculator chips---National Semiconductor has a new scientific calculator chip (the MM57109) that uses RPN formatted problem entry, a 4 level stack, and has a parallel input/output scheme. (perfect for hooking to your micro)

The 57109 (around $18.00) would need about 6 or 7 chips to interface to your machine, but the software driver would be minimal. I have one of these devices and will be hooking it up when I get the time.

The RPN calculator freaks are going to love this one!!!

Speaking of RPN calculators (I could'nt resist that one)--- Popular Electronics (June 1977) presented about 6 game programs written for the HP-25 that could be adapted to KIM.

Who'll be the first to develop a universal game board interface for KIM??? Plasma or liquid crystal would make dandy display panels but may still be a bit cost prohibitive at this time. Maybe a slew of those three digit miniature 7-segment could be tied together, or something along those lines. Any ideas????????

HIGH-SPEED MASS STORAGE

As I see it, there are three distinct possibilities for high-speed mass storage for KIM at this time. The Digital Group dual Phi-deck system, the 3M3A National Multiplex cartridge deck, and a floppy disc.

If you are working with any of these storage mediums, I would be interested in hearing from you.

Since the software will present the biggest the biggest hassle, it would be most efficient to work along with several others who are into the same thing.

I would consider devoting a whole issue to the proper file-handling software for these type devices and I feel certain that the rest of the 6502 fraternity would be most appreciative of your efforts.

until next time.....